

## Summary Praat: Controlling the User - Pauses <sup>1</sup>

### 1. Controlling the user

You can temporarily halt a Praat script

**pause** *text* suspends execution of the script, and allows the user to interrupt it; a message window will appear with the text and the buttons *Stop* and *Continue*

**pause** The next file will be beerbeet.TextGrid

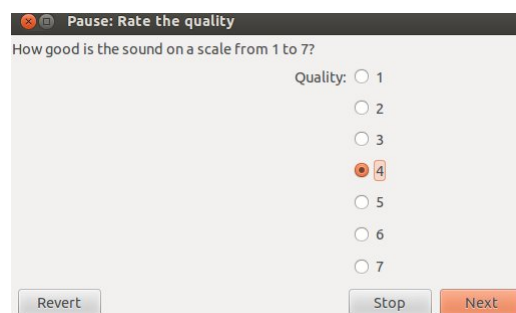


You can also include the same kinds of arguments as for a form. But to do so, you have to use a different syntax!

```

1  for i to 5
2      Read from file: "sound" + string$ (i) + ".wav"
3      Play
4      Remove
5      beginPause: "Rate the quality"
6          comment: "How good is the sound on a scale from 1 to 7?"
7          choice: "Quality", 4
8              option: "1"
9              option: "2"
10             option: "3"
11             option: "4"
12             option: "5"
13             option: "6"
14             option: "7"
15      endPause: if i = 5 then "Finish" else "Next" fi, 1
16      appendInfoLine: quality
17  endfor

```



**rows 5-15** use the "new" syntax as already known from our other scripts, this is different to the syntax of the **forms**

<sup>1</sup>Reference: Praat User Manual - Scripting 6.6 Controlling the user [13.06.2014]

**row 15** if *i* equals 5 the button to proceed will change from **Next** to **Finish**; the end of the argument *fi* equals the `endif` command used within a script; 1 defines the button that is highlighted

### File selection

If you want the user to choose a file name for reading (opening), use the following...

```

1  fileName$ = chooseReadFile$: "Open a table file"
2  if fileName$ <> ""
3      table = Read Table from tab-separated file:  fileName$
4  endif

```

A window will appear and the user will be able to choose a (here:) table file. If the user clicks **OK**, the variable *fileName\$* will contain the name of the file that the user selected. If the user clicks **Cancel**, the variable will contain an empty string ("").

Do the following if you want the user to choose a file name for writing (saving)...

```

1  selectObject: "Sound example"
2  fileName$ = chooseWriteFile$: "Save as a WAV file", "example.wav"
3  if fileName$ <> ""
4      Save as WAV file:  fileName$
5  endif

```

A window will appear and the user will be able to choose an appropriate file name (in this example a file name has been suggested: *example.wav*). If the user clicks **OK**, the form will ask for confirmation if the file name that the user typed already exists. If the user clicks **OK** with a new file name, or clicks **OK** in the confirmation window, the variable *fileName\$* will contain the file name that the user typed; if the user clicks **Cancel** at any point, the variable *fileName\$* will contain the empty string ("").

If you want the user to choose a directory (folder) name, do the following...

```

1  directoryName$ = chooseDirectory$: "Choose a directory to save all the new files in"
2  if directoryName$ <> ""
3      for i to numberOfSelectedSounds
4          selectObject:  sound [i]
5          Save as WAV file:  directoryName$ + "/sound" + string$ (i) + ".wav"
6      endfor
7  endif

```

A window will appear and the user will be able to choose a directory. If the user clicks **OK**, the variable *directoryName\$* will contain the name of the directory that the user selected; if the user clicks **Cancel**, the variable *directoryName\$* will contain the empty string ("").

## 2. Exercise

Write a script that...

- ...reads in all sound files of a directory
- ...creates a TextGrid for each WAV file
- ...pauses the script so the user can segment and annotate within the TextGrid
- ...continues when the user finished annotating
- ...saves the TextGrid and checks whether a TextGrid with the same name already exists (see `fileReadable`); if it does, pause the script to rename the object